



PUD

| | |
|---|---------------------------------------|
| Disciplina: COMPILADORES | Código da Disciplina: CCMP0051 |
| Carga horária total: 60 h Carga horária teórica: 60 h Carga horária prática: 0 h | Período de oferta: 2012.2 |
| | Turma: C7 |
| Professor responsável: MARCUS VINÍCIUS MIDENA RAMOS | |

EMENTA:

Linguagens-fonte, objeto, de alto-nível e de baixo-nível. Especificação de linguagens de programação. Compilação e interpretação. Processadores de linguagens de programação. Máquinas reais e virtuais. Bootstrapping. Análise sintática. Análise de contexto. Ambientes de execução. Geração de código. Otimização de código independente de máquina. Otimização de código dependente de máquina.

CALENDÁRIO:

| Aula | Assunto | Dia | Carga |
|------|--|-------|-------|
| 01 | Motivação Linguagens de alto e baixo-nível Linguagens fonte e objeto Compilação e interpretação Tipos de processadores Métodos para especificação de linguagens | 13/12 | 2hs |
| 02 | Processadores de linguagens Diagramas-T Representação de processos | 18/12 | 2hs |
| 03 | Compilador portátil Bootstrapping Desenvolvimento incremental de linguagens | 20/12 | 2hs |
| 04 | Especificação de linguagens Revisão de gramáticas livres de contexto Sintaxe livre de contexto da linguagem mini-Triangle | 22/01 | 2hs |
| 05 | Sintaxe livre de contexto x sintaxe dependente de contexto | 24/01 | 2hs |



| | | | |
|----|--|-------|-----|
| | Sintaxe dependente de contexto da linguagem mini-Triangle Semântica da linguagem mini-Triangle Fases de compilação Front-end e back-end | | |
| 06 | Exemplo em mini-Triangle: programa-fonte, AST, AST decorada e programa-objeto Passos de compilação Um passo e vários passos: vantagens e desvantagens | 05/02 | 2hs |
| 07 | Estrutura léxica Símbolos úteis e inúteis Tokens, tipo e valor Alfabeto1 -> linguagem1 = alfabeto2 -> linguagem2 Expressões regulares, BNF e EBNF | 07/02 | 2hs |
| 08 | Substituição de não-terminais Fatoração à esquerda Eliminação de recursão à esquerda Starter Reduções e derivações Análise sintática descendente e ascendente | 14/02 | 2hs |
| 09 | Gramáticas e linguagens LL(k) e LR(k) Definição de gramática LL(1) Gramáticas LL(1) sem regras vazias Gramáticas LL(1) com regras vazias | 19/02 | 2hs |
| 10 | Exemplos e exercícios Gramática léxica Gramática sintática | 21/02 | 2hs |
| 11 | Método recursivo descendente Aplicação do método recursivo descendente à construção de analisadores sintáticos | 23/02 | 2hs |
| 12 | Aplicação do método recursivo descendente à construção de analisadores léxicos Aspectos de implementação | 26/02 | 2hs |
| 13 | Desenvolvimento do projeto e esclarecimento de dúvidas | 28/02 | 2hs |
| 14 | Montagem da AST Análise de contexto: identificação Tabela de identificação: atributos e métodos | 05/03 | 2hs |



| | | | |
|----|--|-------|-----|
| | Linguagens com estrutura de bloco monolítica | | |
| 15 | Prova 1 | 07/03 | 2hs |
| 16 | Linguagens com estrutura de blocos planos Linguagens com estrutura de blocos aninhados Escopo estático x escopo dinâmico | 12/03 | 2hs |
| 17 | Verificação de tipos Tipo estático x tipo dinâmico Implementação de analisadores de contexto Padrão de projeto Visitor | 14/03 | 2hs |
| 18 | Representação de dados Tipos simples Tipos agregados heterogêneos | 19/03 | 2hs |
| 19 | Tipos agregados homogêneos Verificação de índices Origem virtual Múltiplas dimensões | 21/03 | 2hs |
| 20 | Exemplos de representação de dados Avaliação de expressões em máquinas de registradores Avaliação de expressões em máquinas de pilha Instruções LOAD, LOADL, STORE e CALL | 23/03 | 2hs |
| 21 | Alocação de memória - estática Alocação de memória - automática Estrutura do stack-frame (SB, LB, ST, L1, L2, ..., LD, LE, ER) | 26/03 | 2hs |
| 22 | Passagem de parâmetros Retorno de valor de função Exemplo completo | 02/04 | 2hs |
| 23 | Instruções CALL, RETURN, LOADA, LOADI, STOREI, PUSH e POP Passagem de parâmetros por endereço x por valor Exemplo completo | 04/04 | 2hs |
| 24 | Cálculo dos registradores de acesso às variáveis e do link estático Funções de código Padrões de código Exemplos de padrões de código Instruções JUMP, JUMPIF | 09/04 | 2hs |



| | | | |
|----|--|-------|-----|
| 25 | Exemplos de aplicação de funções e padrões de código Implementação dos métodos de geração de código | 11/04 | 2hs |
| 26 | Padrão de código para declaração de procedimentos e funções, comandos for e switch; Organização da memória, pilha e heap. | 16/04 | 2hs |
| 27 | Revisão Visitor Exemplo funcional completo com análise sintática, impressão da árvore de sintaxe, análise de contexto e geração de código (códigos fonte e objeto). | 18/04 | 2hs |
| 28 | Orientação de projeto | 20/04 | 2hs |
| 29 | Conclusões: Ciclo de vida de linguagens de programação Detecção e recuperação de erros Eficiência da execução Interpretores Geradores de compiladores | 23/04 | 2hs |
| 30 | Prova 2 | 25/04 | 2hs |
| | Segunda chamada | 30/04 | 2hs |
| | Prova final | 02/05 | 2hs |

| | |
|------------------|-------------------------------------|
| Carimbo do DRCA: | Emitido pelo DRCA em ____/____/____ |
|------------------|-------------------------------------|